

Département informatique

CAHIER DES CHARGES

Version 0.1



Robot télécommandé

Projet d'architecture des systèmes de première année

Enseignants responsables et clients :

Auguste LE VAN SUU
François TOUCHARD

** Le présent document devra être signé à réception par l'un des clients, au moins.*

Maîtres d'œuvres :
Jonathan FAVIER
Sylvain LARGIER

Promotion 2008

Jeudi 16 février 2006

SOMMAIRE

I. Introduction

1.1 Objet

L'objet de ce cahier des charges est de décrire les clauses techniques, qualitatives et administratives de la réalisation d'un robot télécommandé. On peut trouver en annexe 1 des précisions sur cette réalisation (à travers l'analyse des besoins remise par le client).

1.2 Contexte

Cette étude se déroule dans le cadre du projet d'architecture proposé aux étudiants de première année du département informatique de l'ESIL. L'équipe de développement, composée de Jonathan FAVIER et Sylvain LARGIER, sera supervisée par M. Auguste LE VAN SUU. L'objectif de ce projet est de nous placer en situation réelle de gestion de projet. Nous devons en effet passer par toutes les étapes de la création d'un projet, de l'étude préalable à la présentation au client – représenté ici par les enseignants de l'ESIL – de la version finale du produit, en passant par l'acquisition de connaissances nécessaires au franchissement de certains obstacles technologiques.

1.3 Terminologie

Vous pouvez trouver à la fin de ce document un glossaire définissant les termes techniques utilisés dans les chapitres qui suivent. La première occurrence de chacun des mots répertoriés dans le glossaire sera soulignée. Une définition des termes techniques utilisés pour la réalisation du robot est également donnée en annexe 1.



II. Description détaillée du produit

2.1 Caractéristiques et exigences fonctionnelles

Le matériel de base sera directement fourni par le fournisseur, M. LE VAN SUU (le robot, la caméra, etc.), et le développement du projet sera réalisé également avec un PC de l'ESIL.

Le projet du robot à pour but de construire et développer un robot complet télécommandé qui peut se déplacer en fonction des ordres données par son utilisateur (déplacements vers l'avant, vers l'arrière, vers la gauche et vers la droite), via une télécommande (qui sera fera sous forme de reconnaissance vocale) et via le PC. Ce robot intégrera donc une antenne réceptrice pour recevoir les commandes, mais également une caméra permettra de visualiser sur le PC le trajet du robot. Sa source d'alimentation sera une pile.

Le boîtier de commande permettra de contrôler les différents mouvements du robot. Il sera mu'une carte d'acquisition de la voix qui permettra d'enregistrer puis d'envoyer les commandes vers le robot. Il contiendra également une antenne émettrice, une prise mini jack pour brancher le micro récepteur (pour la reconnaissance vocale) et un port série pour la connexion avec un ordinateur.

Enfin, en fonction de l'aisance de l'équipe de conception et du temps qu'il restera en fin de projet, nous pourrons intégrer un certain degré d'intelligence pour les déplacements du robot en autonome. Cette dernière fonctionnalité permettra également d'obtenir des informations sur les obstacles rencontrés par le robot sur son trajet, et peut être réalisé à l'aide d'un détecteur d'obstacles.

La vitesse n'est pas une exigence primordial, ni l'aspect visuel (ces deux caractéristiques pourront être travaillées en fin de projet si le temps le permet). Le poids n'est pas non plus primordial, mais le robot doit rester quand même facilement transportable. Par contre, le robot doit être simple d'utilisation et fiable, et doit répondre à toutes les fonctionnalités demandées par le client.

2.2 Fonctionnement

Le robot comprendra les commandes suivantes :

- Avancer
- Stopper
- Reculer
- Tourner à droite
- Tourner à gauche

L'uateur doit prononcer des mots enregistrés auparavant qui correspondent aux commandes du robot. La commande demandé arrive dans le boîtier de commande grâce au

micro intégré, qui envoie des informations au logiciel grâce au port série. Le logiciel renvoie ensuite l'information au boîtier de commande, qui la renvoie au robot grâce à son antenne émettrice. Le robot exécute alors la commande demandé.

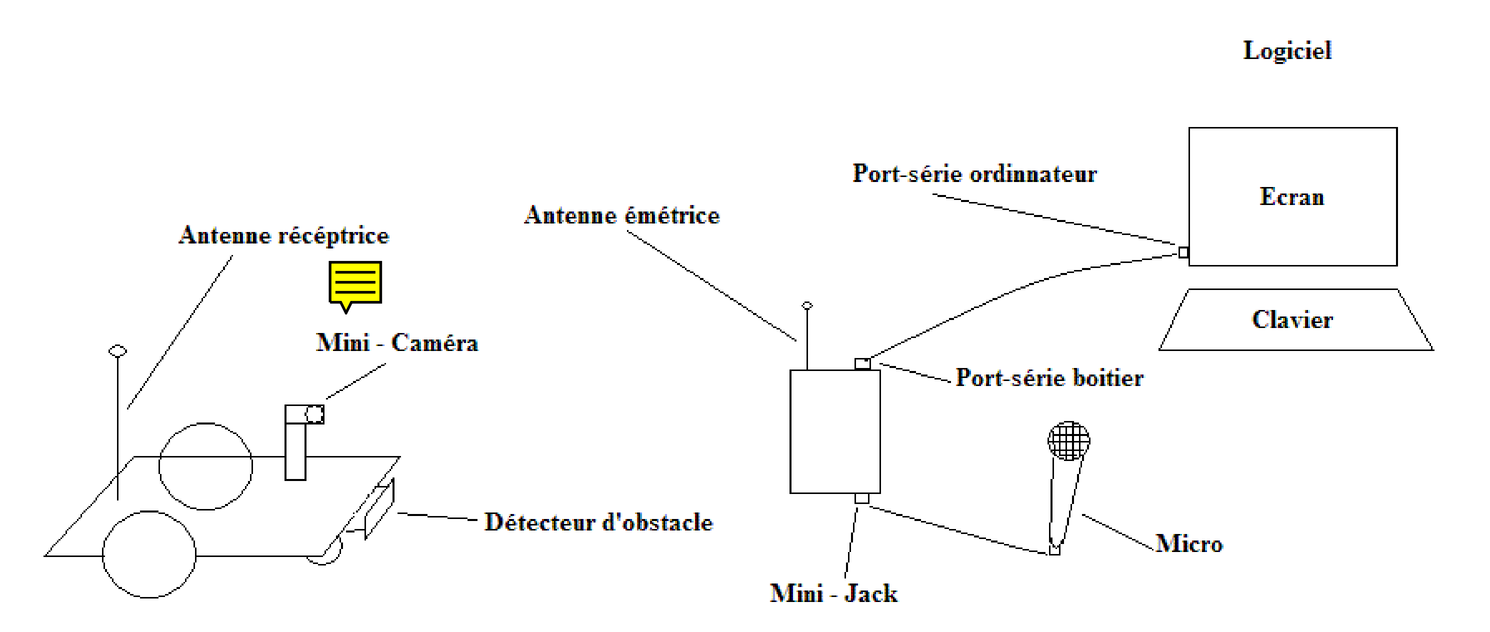
On pourra également contrôler le robot grâce aux touches directionnelles du clavier de l'ordinateur via le logiciel qui y sera installé. Dans ce cas l'information sera directement transmise du logiciel vers le boîtier et du boîtier vers le robot. Ce processus constituera une solution de repli au cas où le système serait compromis (en cas d'erreur dans la reconnaissance de la commande par la carte d'acquisition de la voix).

2.3 Interfaces utilisateur

Un logiciel sera installé dans l'ordinateur et sera capable de passer du mode « reconnaissance vocale » au mode « clavier » et inversement. L'utilisateur aura le choix entre 2 modes de contrôle du robot, qui devront permettre de répondre aux même exigences et qui devront intégrer les mêmes fonctionnalités :

- commandes à travers le PC : nous devrons pouvoir envoyer des ordres au robot via le PC afin de faire déplacer le véhicule en avant, en arrière, vers la gauche et vers la droite. Une commande sera également implantée pour permettre de stopper le véhicule.
- télécommande par reconnaissance vocale : tous les déplacements du PC devront pouvoir être effectués par reconnaissance vocale, à travers des mots enregistrés par l'utilisateur (« Avancer », « Reculer », etc.).

2.4 Synoptique



2.3 Exigences non fonctionnelles

Le fonctionnement du robot devra être fiable. Des phases de tests poussées seront effectuées avant la présentation finale du 3 avril 2006 (jeu d'essai au cours des semaines de développement, séries de tests effectuées dans différents environnements). Le cahier des recettes mettra en relief ces tests ainsi que leur pertinence, et un rapport de tests relatera l'ensemble des tests effectués sur le projet et la manière dont ils seront effectués.



III. Livrables

3.1 Robot et Documentation

Le robot, le CD d'installation du logiciel pour commander le robot sur le PC, et la documentation devront être livrés en même temps et présenté de manière professionnelle.

Par le terme « Documentation », on entend une explication détaillée du fonctionnement technique du logiciel. On y fera apparaître un mode d'emploi pour l'utilisation du robot, un guide d'utilisation pour le logiciel et son installation ainsi que toutes autres informations nécessaires à la maintenance du robot télécommandé. Cette documentation se présentera sous forme papier, et également sous forme de documents PDF présents sur le CD d'installation du logiciel. Elle permettra également de connaître les fonctionnalités demandées au départ par le client, ainsi que la manière dont a été mené le projet.

Pour résumer, la documentation contiendra le cahier des charges (associé également au jeu d'essai, au cahier des recettes, au planning de type PERT et au plan synoptique des modules), le document de préconception, l'analyse fonctionnelle et les documents de tests finaux.

Une documentation utilisateur sera également proposée. Cette dernière rendra compte de toutes les fonctionnalités du robot et du logiciel associé. N'importe quel utilisateur pourra à la lecture de document d'aide maîtriser la totalité des fonctionnalités proposées.

3.2 Garantie et conditions d'application

Le package de livraison comprendra également un livret de garantie contenant les conditions d'applications de la garantie (délais donnant droit à cette garantie, dommages pris en charges, délais de réparation, etc.). Un contact sera mentionné afin de faire valoir cette garantie.

Voici en résumé ce qui composera le packaging du robot télécommandé :

- ✓ Notice technique d'installation.
- ✓ Fiche des caractéristiques techniques.
- ✓ Manuel d'utilisation (documentation utilisateur) du robot.
- ✓ CD-ROM d'installation du logiciel.
- ✓ Fiche de garantie et de SAV.

IV. Conditions de réalisation

4.1 Délai de réalisation

Il a été convenu que ce projet donne lieu à un suivi régulier du travail hebdomadaire effectué par l'équipe en charge du projet. En ce sens, les grandes dates du projet doivent être mentionnées et normalisées dans un planning de type PERT, facilement consultable par les superviseurs du projet.

Le présent cahier des charges devra être rendu au plus tard le 28 février 2006. Le projet devra quant à lui être finalisé pour le lundi 3 avril 2006. Une présentation exhaustive du produit sera alors proposée au client.

Parallèlement, il est nécessaire pour le client d'avoir une bonne visibilité des ressources mises en œuvre pour le projet. Les tâches et la gestion des emplois du temps de ces personnes doivent figurer dans le document de suivi du projet.

Note importante: Fort d'une certaine expérience, le client mise sur une normalisation particulière des contacts qu'il peut avoir avec ses fournisseurs d'applications. La partie « suivi hebdomadaire » présente les règles préconisées pour la transmission de documents.

4.2 Contraintes de développement

Le développement du microcontrôleur PIC se fera dans le langage assemblage MPASM, avec l'outil de développement MPLAB et le programmateur PICSTART. Nous établirons donc des règles de codage pour tous les fichiers sources, de manière à faciliter la lecture du code et à le simplifier. D'autre part le logiciel et les interfaces permettant de contrôler le robot à partir du PC seront codés en JAVA ou en Visual Basic. Des règles de codage pour ce langage seront donc aussi établis.



4.3 Analyse des risques

Bien qu'une soigneuse planification ait été effectuée lors de la réalisation du présent document, de nombreux problèmes inattendus peuvent survenir et retarder le développement du projet. Voilà pourquoi nous présentons un tableau des risques recensant, de la manière la plus exhaustive possible, les risques qui entourent le déroulement de ce projet et sa finalisation. Nous pourrons ainsi mieux anticiper les problèmes afin de mieux les gérer.

Risques	Mesures de prévention
Difficulté à concevoir une partie du projet, notamment la partie « commande vocale » qui s 'annonce la plus complexe	En cas de difficulté de conception, nous en informerons immédiatement notre tuteur pour en « discuter ».
La diffusion des ondes hertziennes peut vite s'avérer problématique en présence d'ondes « parasites »	Le tests sur le module «Réception des ondes hertzienne » se fera dans un lieu clos à l'abri d'ondes parasites
Un composant peut être endommagé ou perdu	La gestion et l'utilisation des composants se fera avec attention
Indisponibilité du client lors des rendez-vous hebdomadaires.	Prise régulière de rendez-vous en dehors des heures de cours.
Problèmes d'utilisations rencontrés par le client	Mise en place d'une hotline 24h/24 au 06.70.75.41.45
Perte des données présentes sur l'ordinateur utilisé pour le développement du logiciel.	Réalisation de sauvegardes quotidiennes sur support amovible.
Indisponibilité exceptionnelle de l'un des membres de l'équipe de développement lors des phases de travail commun.	Organisation d'une réunion extraordinaire en dehors des heures de travail et mise en place d'un plan d'urgence visant une nouvelle synchronisation sur le planning.

4.4 Communication avec le client et suivi de la réalisation

Le principal moyen de communication entre le maître d'œuvre et d'ouvrage sera le courrier électronique. Tous les suivis et documents devront être transmis aux enseignants responsables du projet. Un compte-rendu hebdomadaire sous forme de courrier électronique sera envoyé au client en précisant le travail réalisé et les tâches prévues pour la séance suivante.

Une présentation du produit, de ses fonctionnalités et de ses contraintes d'utilisation aura lieu le 03 avril 2006, comme demandé par le client lors de la remise du projet.

Afin d'assurer à nos clients le meilleur confort d'utilisation, plusieurs services d'assistance seront mis à sa disposition dés la mise sur le marché du produit :

- Garantie « satisfait ou remboursé ».
- Installation et dépannage gratuits à domicile.
- Support technique en ligne.

Lors de la phase de développement du produit, une réunion hebdomadaire avec le client (représenté par Auguste LE VAN SUU) sera organisée.

En cas de problème ou pour toutes questions relatives au projet, les personnes à contacter sont les développeurs Jonathan FAVIER et Sylvain LARGIER :

	Jonathan FAVIER	Robin HAIDER
Fonction	Maître d'œuvre, Membre de l'équipe de développement.	Maître d'œuvre, Membre de l'équipe de développement.
Email	jfavier@esil.univ-mrs.fr	slargier@esil.univ-mrs.fr
Tél.	06.70.75.41.45	06.11.11.05.97

4.5 Suivi du projet

Le contenu du document de suivi du projet est normalisé. Le détail de la planification est présenté dans le diagramme de PERT ci-dessous :

Diagramme PERT

A*. Etudes du système et de l'environnement (4 heures)
A. Programmer le microcontrôleur (8)
B. Module d'acquisition vidéo (4)
C. Module d'acquisition audio (programmer la reconnaissance vocale) (4)
D. Implémentation de l'interface du logiciel (4)
E. Réception des commandes sur le PC (4)
F. Traitement des commandes (6)
G. Emission des ondes « radios » (5)
H. Phases de tests (8)

V. Glossaire

Dans ce document, la première occurrence des termes qui sont définis ci-dessous est soulignée afin d'en faciliter la compréhension.

- **Microcontrôleur :** Circuit intégré rassemblant un microprocesseur et d'autres composants tels que de la mémoire et des périphériques. Les microcontrôleurs améliorent l'intégration et le coût (lié à la conception et à la réalisation) d'un système à base de microprocesseur en rassemblant ces éléments essentiels dans un seul circuit intégré.

- **Radiofréquence :** Zone à haute fréquence du spectre électromagnétique utilisée, entre autre, pour la transmission des signaux de télévision et de radio.

- **Reconnaissance vocale :** La reconnaissance vocale est une technologie informatique permettant à un logiciel d'interpréter une langue naturelle humaine. Le principe est simple : un enregistrement de quelques mots prononcés par un locuteur est interprété en texte. Cela permet entre autres la transcription automatique et le contrôle de systèmes par la voix. Cette technologie utilise des méthodes informatiques des domaines du traitement du signal et de l'intelligence artificielle.

- **Port série :** permet la communication entre deux support qui y sont équipés

VI. Annexes

6.1 Annexe 1 : analyse des besoins du client

Extrait de l'analyse des besoins livré par le client:

Le projet de robot consiste à construire et développer un système robot complet télécommandé intégrant une reconnaissance vocale ainsi qu'un embryon d'acquisition d'images vidéo.

L'élément central du projet est un robot (au départ en kit…) possédant un système autonome de contrôle pour les déplacements. Nous devrons développer un système permettant de télécommander ce robot. Pour cela nous devrons maîtriser les étapes de développements mécaniques électriques et de programmations logicielles de tous les éléments qui vont intervenir dans ce système.

Cette télécommande se fera si possible par reconnaissance vocale. Suivant notre aisance nous ajouterons un certain degré d'intelligence pour les déplacements du robot en autonome. De manière annexe le robot devra aussi donner des informations d'états sur les évènements qu'il peut rencontrer sur son trajet (choc sur un obstacle).

6.2 Annexe 2 : Conventions de programmation

Voici comment se présentera l'entête des différents fichiers sources réalisés :

```
/**
*
* @File: tp_01.xx
*
* @Authors: S.LARGIER et J.FAVIER
*
* @Date : 18/02/06
* @Version : 1.0
*
* @Synopsis : Exemple d'entête de fichier qui
* peut se continuer sur plusieurs lignes.
*
* @Bugs : indiquer les erreurs connues et non encore corrigées
*
* @ToDo: indiquer ce qui reste a faire
*
* @Modified:
* @@Authors: S. LARGIER
* @@Date : 18/02/06
* @@Synopsis : correction de la date de création...
*
* @Modified:
* @@Authors: J. FAVIER
* @@Date : 26/02/06
* @@Synopsis : correction de la correction de M. Dupont...
*
**/
```

Le contenu du fichier devra ressembler à ceci :

```
/*-------------- Inclusions des headers nécessaires ------------*
includes systeme et headers créés pour le projet

/*-------------- Définitions des macro-instructions ------------*
Toutes les commandes #define (pas de double emploi avec l'include correspondant. Cette
remarque s'applique également à ce qui suit)

/*-------------- Définitions des types étiquetés ---------------*
Déclarations de types étiquetés (enum, structures, unions)
Définitions de types synonymes à l'aide de typedef

/*-------------- Déclarations des variables globales -----------*

/*-------------- Déclarations des fonctions --------------------*/

/*****************************************************************
*****           Fonctions publiques du module               *****
*****************************************************************/

PUBLIC type_du_résultat nom_de_la_fonction(liste de paramètres)
/*---------------------------------------------------------------
* Description    :
* Entrées        :
* Sorties        :
* Code exception :
---------------------------------------------------------------*
{
Déclarations par référence de type IMPORT (non recommandé pour les fonctions)
Déclarations internes
Instructions du corps de la fonction
}

/*****************************************************************
*****           Fonctions privées du module                 *****
*****************************************************************/
PRIVATE type_du_résultat nom_de_la_fonction(liste de paramètres)
```

Voici quelques conventions concernant le nom des identificateurs du programme :

- **Identificateurs des variables :** Chaque mot composant l'identificateur commence par une majuscule, à l'exception du premier, et ne contient que des lettres et des chiffres (exemple : maVariable).

- **Identificateurs des paramètres formels :** ces identificateurs respectent la même syntaxe que les variables, en y ajoutant au début "p_".

- **Identificateurs des fonctions :** les identificateurs des fonctions respectent la même syntaxe que ceux des variables à la différence que le premier caractère est systématiquement en majuscule au lieu d'être en minuscule.

- **Identificateurs des prédicats ou variables booléennes :** la lisibilité des identificateurs des prédicats sera améliorée en les préfixant de « Est ». Par exemple, l'appel à la fonction `EstFeuille()` permettra de savoir si le nœud passé en paramètre est une feuille.

- **Identificateurs des constantes symboliques :** les constantes symboliques seront identifiées avec des majuscules (exemple : PUBLIC, TAILLE_CHAR, etc.).